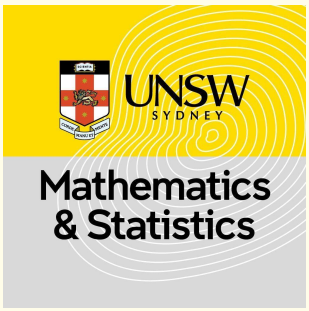


MATH5805
A Modern Introduction to Extreme Value Theory
Term 1, 2025

Dr. Boris Béranger

Week 9 - Lab Solutions

Exercise 1

In this exercise we explore how to fit multivariate point process models via the approximate likelihood. For simplicitly, let's use simulated data.

a) Generate 5000 samples from a bivariate $t_\nu(0, \Sigma)$ distribution (`R` package `mvtnorm`), with $\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ for $\rho = 0.5$, and $\nu = 3$ degrees of freedom. Plot the data before and after transformation to the unit Fréchet scale. *Hint:you might want to use the standardisation functions written in the Week 7 Lab.*

Solution:

```
x2u <- function(x){
 lx <- nrow(x)
 p <- ncol(x)
 u <- matrix(NA, nrow=lx, ncol=p)
 for(i in 1:p){
   u[,i] <- rank(x[,i], ties.method="random", na.last="keep")/(lx+1)
 }
 u
}

u2f <- function(u){
  lu <- nrow(u)
  p <- ncol(u)
  f <- matrix(NA, nrow=lu, ncol=p)
  for(i in 1:p){
    f[,i] <- -1/log(u[,i])
  }
  f
}

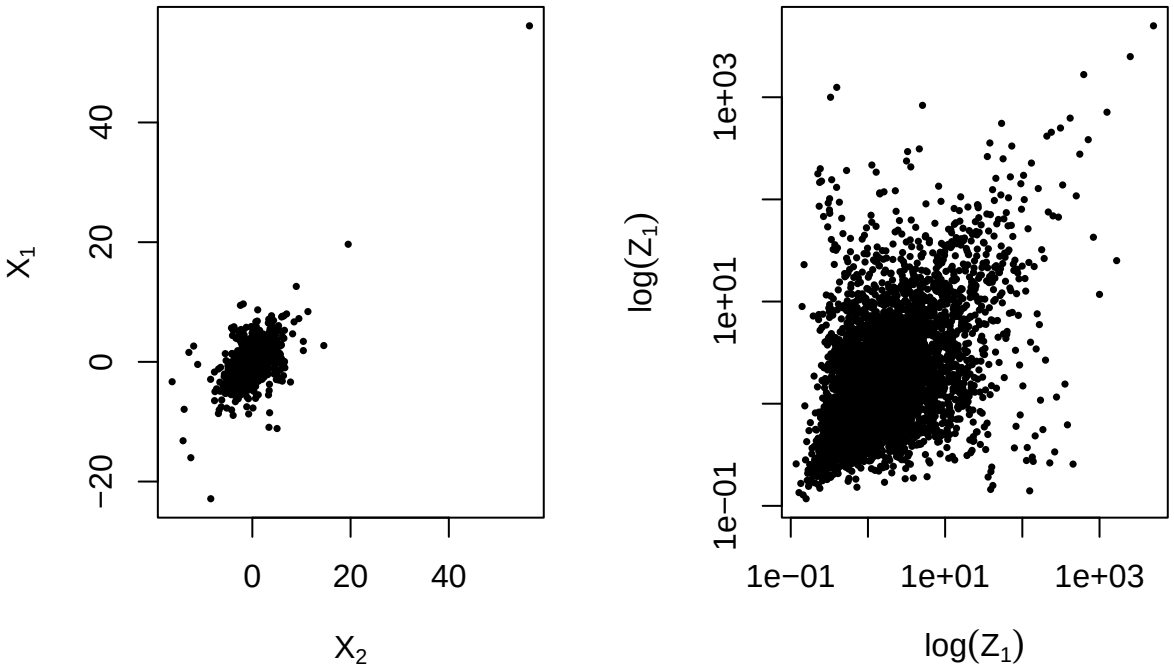
x2f <- function(x){
  u2f(x2u(x))
```

```
}

library(mvtnorm) # For mv normal/t
L <- 5000 # number of samples
rho <- 0.5
set.seed(123)
dat <- rmvt(L,sigma=matrix(c(1,rho,rho,1),ncol=2),df=3)

par(mfrow=c(1,2))
plot(dat,pch=16,cex=0.5,ylab=expression(X[1]),xlab=expression(X[2]))

dat <- x2f(dat) # Transform to unit frechet margins
plot(dat,pch=16,cex=0.5,log="xy",ylab=expression(log(Z[1])),
     xlab=expression(log(Z[1])))
```



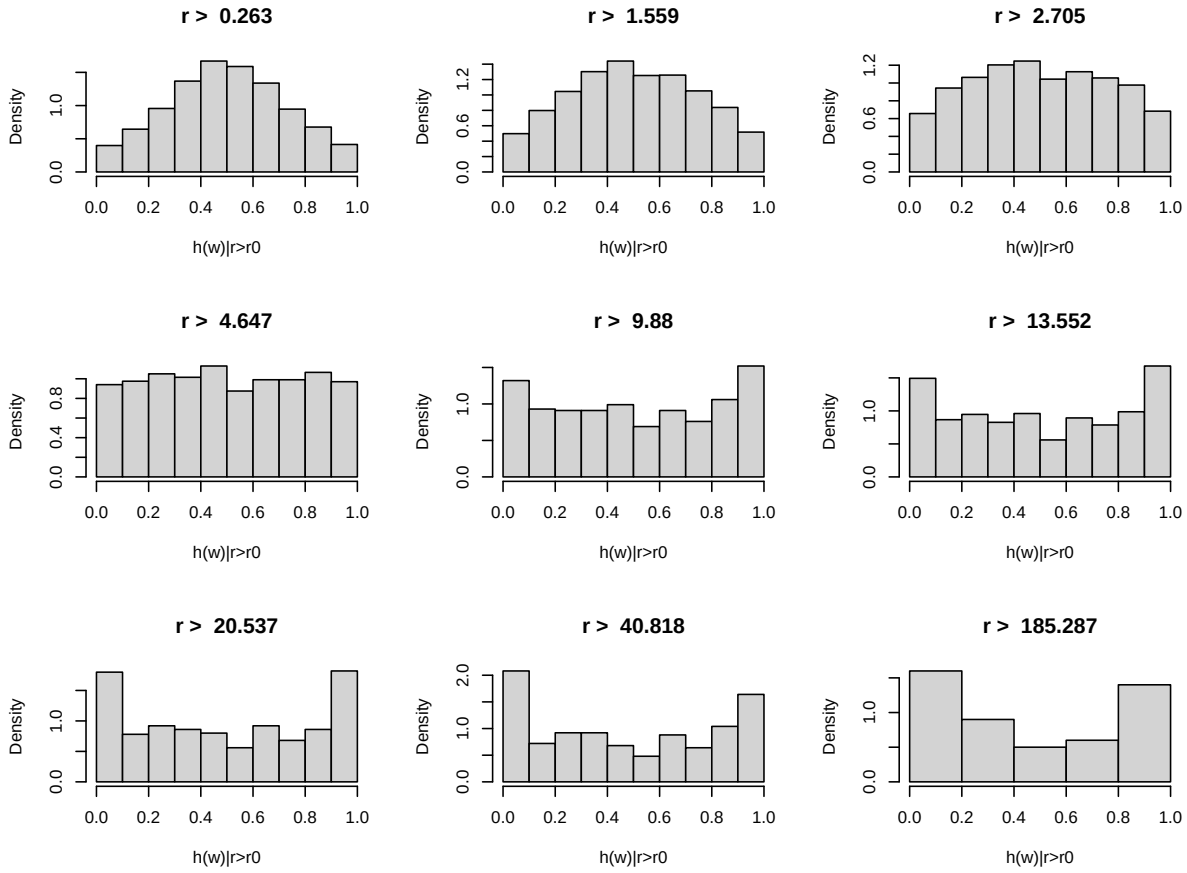
Weak dependence is evident ($\rho = 0.5$).

b) Transform the (X_1, X_2) co-ordinates to the pseudo-polar co-ordinate representation (r, w). In order to use the approximate likelihood we need to identify a value, r_0, such that $w \perp r \mid r > r_0$. Produce histograms of $w \mid r > r_0$ for r_0 defined by the observed 0 (minimum), 0.2, 0.4, 0.6, 0.8, 0.85, 0.9, 0.95, and 0.99 quantiles of r. Comment on your findings.

Solution:

```
r <- rowSums(dat)
w <- dat[,1]/r
q <- c(0,0.2,0.4,0.6,0.8,0.85,0.9,0.95,0.99)
r0 <- quantile(r,q)
par(mfrow=c(3,3))
```

```
for (i in 1:length(r0))
  hist(w[r>r0[i]], main=paste("r > ",round(r0[i],3)),
       probability=T, xlab="h(w)|r>r0")
```



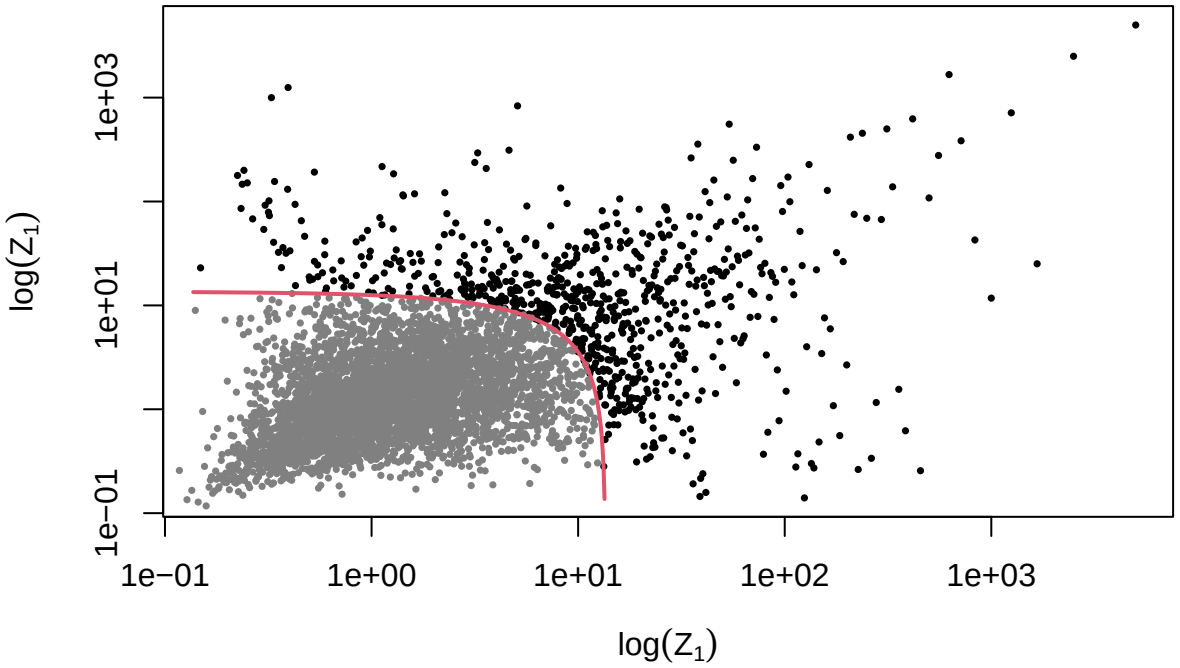
As r_0 increases, the shape of the dependence changes from clear positive dependence for low r_0 to weak dependence from around $r > 13.663$. This highlights that dependence in the extremes is not necessarily related to dependence in the main body of the data! Note that we also get less data above the threshold, as the threshold increases.

c) Hence choose a suitable threshold, r_0, and superimpose the threshold on a plot of the (Fréchet-scale) data.

Solution:
From the above plots it appears that the histograms are loosely unchanging (i.e. the estimated angular density is independent of the threshold) from around the 5th or 6th plot upwards. Hence, taking a value of $r_0 = 9.88$ or $r_0 = 13.552$ or above would appear to be suitable. For the following we take $r_0 = 13.552$.

```
i <- 6 # choose threshold
ind <- r>r0[i]
# par(mfrow=c(1,1))
plot(dat[ind,1], dat[ind,2], pch=16, cex=0.5, log="xy",
     ylab=expression(log(Z[1])), xlab=expression(log(Z[1])))
points(dat[!ind,1], dat[!ind,2], pch=16, cex=0.5, col=grey(0.5))
```

```
xx <- seq(0, r0[i], length=100)
lines(xx, r0[i]-xx, col=2, lwd=2)
```



As can be seen, the threshold has a particular shape (on the log-log scale).

d) Using the `fExtDep(method="PPP", ...)` function in `ExtremalDep` and the angular measurements above the threshold $w \mid r > r_0$, fit the extremal-t and Hüsler-Reiss models. Superimpose the fitted angular densities (check the function `dExtDep()`) on the empirical estimate, and assess the fit. (Note that the extremal-t density needs to choose a small value c such that $w < c$ and $w > 1 - c$ is added to the mass on the vertices. Try using $c = 0.01$ here, but you can explore using different values.

Solution:

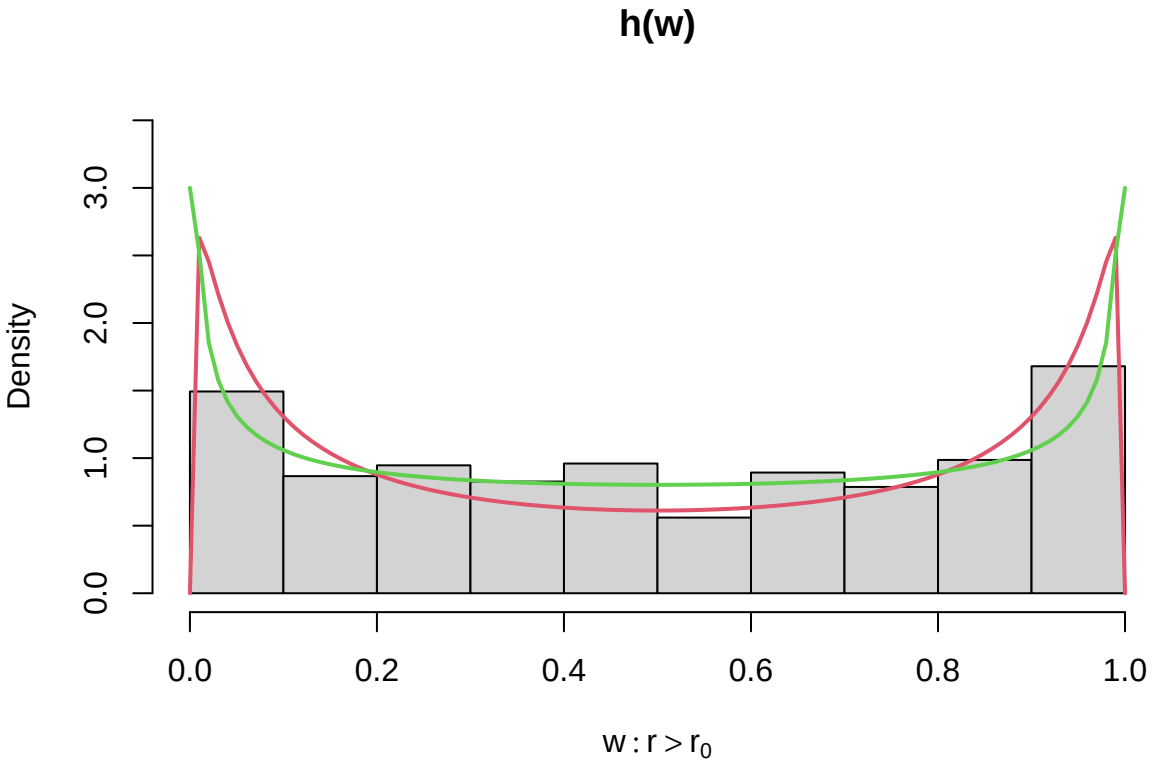
```
library(ExtremalDep)
ddat <- w[r>r0[i]]
ddat <- cbind(ddat,1-ddat)
fit.et <- fExtDep(model="ET", data=ddat, par.start=c(0.5,2), c=0.01, trace=T)
```

```
## initial  value -488.682608
## final  value -561.467674
## converged
```

```
fit.hr <- fExtDep(model="HR", data=ddat, par.start=1, trace=T)
```

```
## initial  value -507.405798
## final  value -536.126432
## converged
```

```
hist(w[r>r0[i]], probability=T, ylim=c(0,3.5), main="h(w)",
     xlab=expression(w : r>r[0]))
xx <- simplex(n=100, d=2)
dhr = apply(xx, 1, dExtDep, model="HR", par=fit.hr$par)/2
det = apply(xx, 1, dExtDep, model="ET", par=fit.et$par, c=0.01)/2
lines(xx[,1], dhr, col=2, lwd=2)
lines(xx[,1], det, col=3, lwd=2)
```



Both models follow the general ∪-shape of the empirical spectral density estimate. The Hüsler-Reiss model (red) visually exhibits a slightly lower dip for central angles than the extremal-t, the latter of which seems to have a more uniform fit. Note that the extremal-t has explicit mass on $\{0\}$ and $\{1\}$ (here expressed by $\omega < 0.05, 1-\omega > 0.95$), unlike the Hüsler-Reiss model, and so the latter appears to follow the empirical density better in this region. (Comment: We had to divide the density by $d = 2$ so that it integrates to 1.)

e) Compute and interpret the extremal coefficient and the coefficient of upper tail dependence for both models. Do the estimates make sense?

Solution:

```
theta.hr <- index.ExtDep(object="extremal", model="HR", par=fit.hr$par)
theta.hr
```

```
## [1] 1.622764
```

```
2-theta.hr
```

```
## [1] 0.3772363
```

```
index.ExtDep(object="upper.tail", model="HR", par=fit.hr$par, x=c(1,1))
```

```
## [1] 0.3772363
```

```
theta.et <- index.ExtDep(object="extremal", model="ET", par=fit.et$par)
theta.et
```

```
## [1] 1.618984
```

```
2-theta.et
```

```
## [1] 0.3810164
```

```
index.ExtDep(object="upper.tail", model="ET", par=fit.et$par, x=c(1,1))
```

```
## [1] 0.3810164
```

The extremal coefficient θ can be interpreted as the number of independent components. So here we know that $1 \leq \theta \leq 2$. Both models estimate θ at around 1.6. This makes intuitive sense as we have only weak dependence ($\rho = 0.5$) between our variables, so we have close to 2 independent pieces of information.
The coefficient of upper tail dependence χ is the probability that all components are simultaneously extreme (where "extreme" here means > 1). The larger χ is, the stronger the tail dependence. Here we have χ estimated around 0.3-0.4, which is not high (strong dependence would give us values closer to 1), but it is also not low (independence would give us values closer to 0). So we have moderate-to-low dependence here, which makes sense.

f) Under each model, calculate the return period for return levels in the range $(r_0, 250)$ for one univariate margin Z_1 (on the scale of numbers of observations). Now repeat this by conditioning on the other univariate margin $Z_2 > r_0$ being greater than the threshold, r_0. (Hint: $\mathbb{P}(A \mid B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$.) Construct and interpret return period/level plots. Do your calculations make intuitive sense?

Solution:

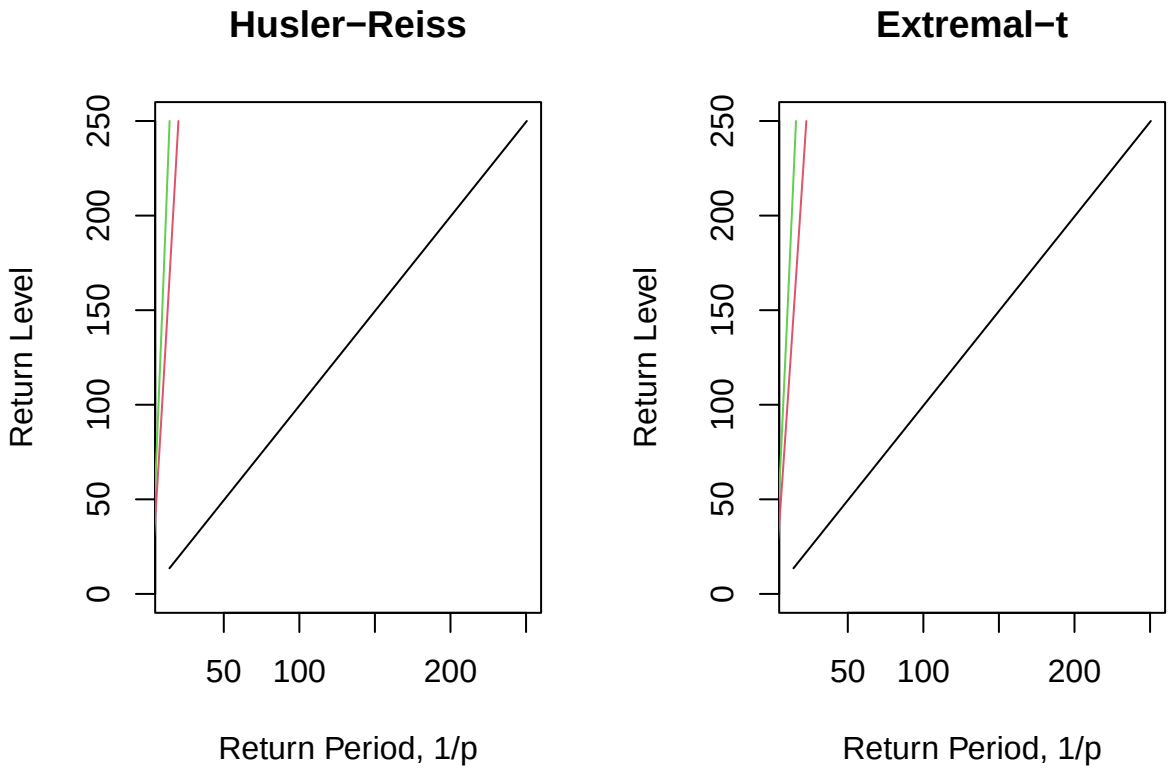
```
zz <- seq(r0[i], 250, length=250)
rp.marg <- 1/pGEV(zz, loc=1, scale=1, shape=1, lower.tail=F)

par(mfrow=c(1,2))
plot(rp.marg, zz, type='l', ylim=c(0,max(zz)), xlab="Return Period, 1/p",
     ylab="Return Level", main="Husler-Reiss")
zzz <- cbind(zz, rep(r0[i],length(zz)))
a <- apply(zzz, 1, pExtDep, model="HR", par=fit.hr$par, type="upper")
p1 <- pGEV(r0[i], loc=1, scale=1, shape=1, lower.tail=F)
lines(1/(a/p1), zz, col=2)

zzz <- cbind(zz, rep(r0[i+1], length(zz)))
a <- apply(zzz, 1, pExtDep, model="HR", par=fit.hr$par, type="upper")
p2 <- pGEV(r0[i+1], loc=1, scale=1, shape=1, lower.tail=F)
lines(1/(a/p2), zz, col=3)
```

```
plot(rp.marg, zz, type='l', ylim=c(0,max(zz)), xlab="Return Period, 1/p",
     ylab="Return Level", main="Extremal-t")
zzz <- cbind(zz, rep(r0[i],length(zz)))
a <- apply(zzz, 1, pExtDep, model="ET", par=fit.et$par, type="upper")
lines(1/(a/p1), zz, col=2)

zzz <- cbind(zz, rep(r0[i+1],length(zz)))
a <- apply(zzz, 1, pExtDep, model="ET", par=fit.et$par, type="upper")
lines(1/(a/p2), zz, col=3)
```



Obtaining the univariate return levels is straightforward under the unit Fréchet distribution. For the conditional return levels, we want $p = \mathbb{P}(Z_2 > z \mid Z_1 > r_0) = \mathbb{P}(Z_2 > z \cap Z_1 > r_0)/\mathbb{P}(Z_1 > r_0)$ from which the resulting return period is $1/p$.
For each plot the unconditional univariate return level/period curve is shown in black. The red line is the return period of the conditioned event, which is substantially higher than the unconditioned event. This says that if we know that one margin is high, the chance of the second margin being high increases, and accordingly the return period for a given return level is reduced.
Also shown (in green) is another return period/level line with a higher conditioning threshold.

Exercise 2

Consider again the historical AUD/xxx daily exchange rates data from, given in the file `AUD-exchangeRates.csv`. In Exercise 3 of the Week 8 Lab, we analysed the multivariate extremes of the 3-dimensional log daily returns of the AUD/(USD, CNY, JPY) exchange rates

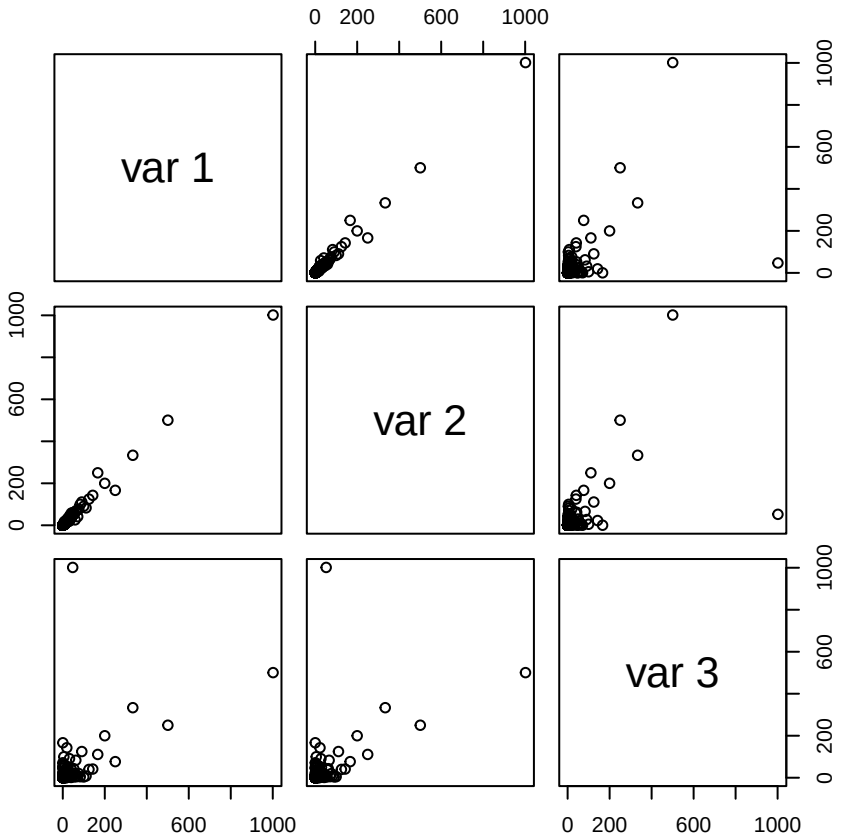
(columns FXRUSD, FXRCR, FXRJY) from a block maxima perspective. Now we will do this from a threshold-exceedance perspective using the approximate likelihood for the point process representation.

a) Read in the data, construct the 3-dimensional log-daily returns, and transform the margins to unit Fréchet (using `x2f`). Examine bivariate scatterplots of the data.

Solution:

```
dat <- read.table("../data/AUD-exchangeRates.csv",header=TRUE,sep=",")
usd <- dat$FXRUSD
cny <- dat$FXRCR
jpy <- dat$FXRJY
L <- length(usd)
ldrusd <- log(usd[2:L]/usd[1:(L-1)])
ldrcny <- log(cny[2:L]/cny[1:(L-1)])
ldrjpy <- log(jpy[2:L]/jpy[1:(L-1)])

dat <- x2f(cbind(ldrusd, ldrcny, ldrjpy))
pairs(dat)
```



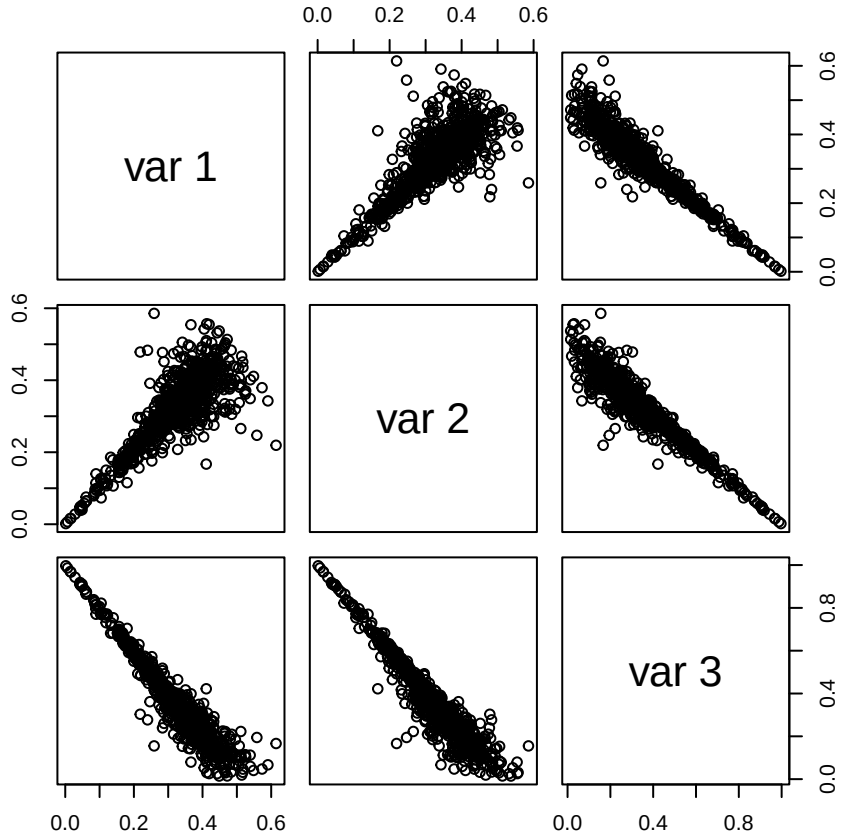
The strong positive relationships between the exchange rates (particularly USD and CNY) are apparent.

b) Transform the data to pseudo-polar co-ordinates (r, ω), and examine the bivariate scatterplots

of the angular component ω.

Solution:

```
r <- apply(dat, 1, sum)
w <- dat/r
pairs(w)
```



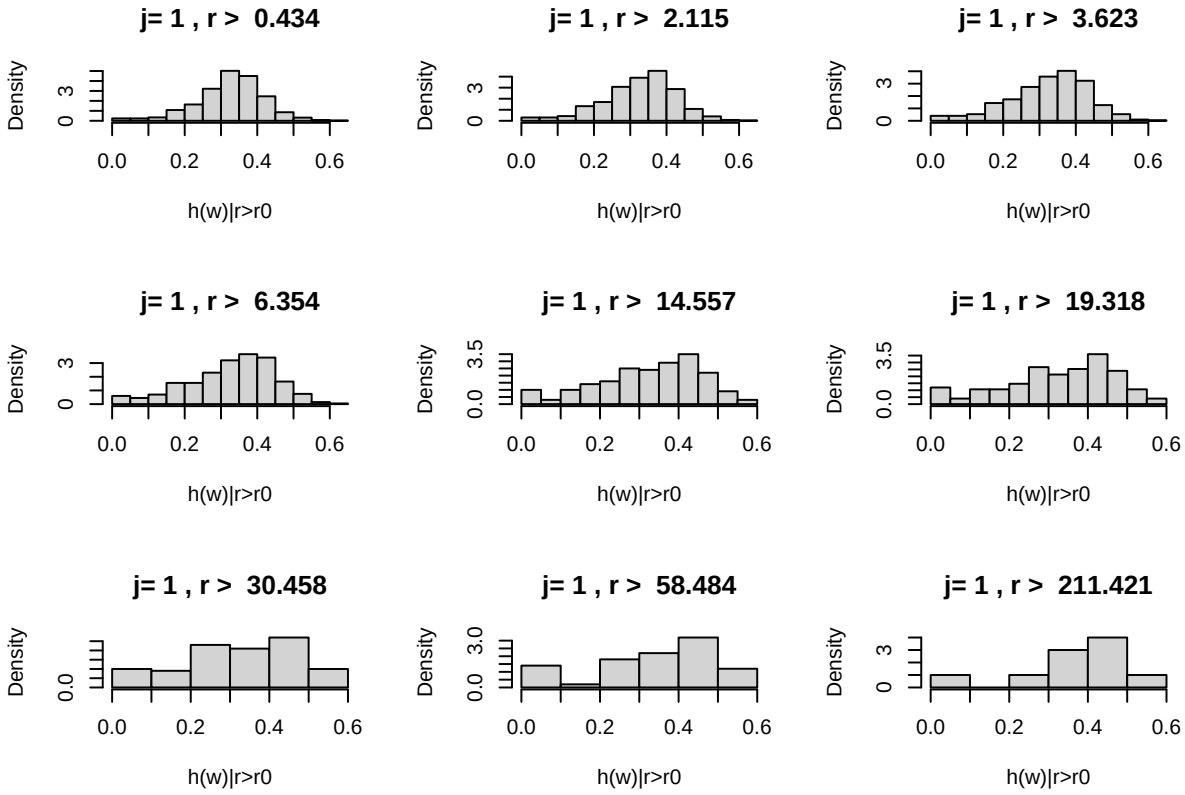
The angular component relationships between the variables are different to the original data due to the sum-to-one constraint. Hence, while for the two highly correlated exchange rates (Var 1, Var 2) when one variable is large the other also tends to be large, the larger they both are, the smaller the 3rd exchange rate (Var 3) has to be, due to the constraint.

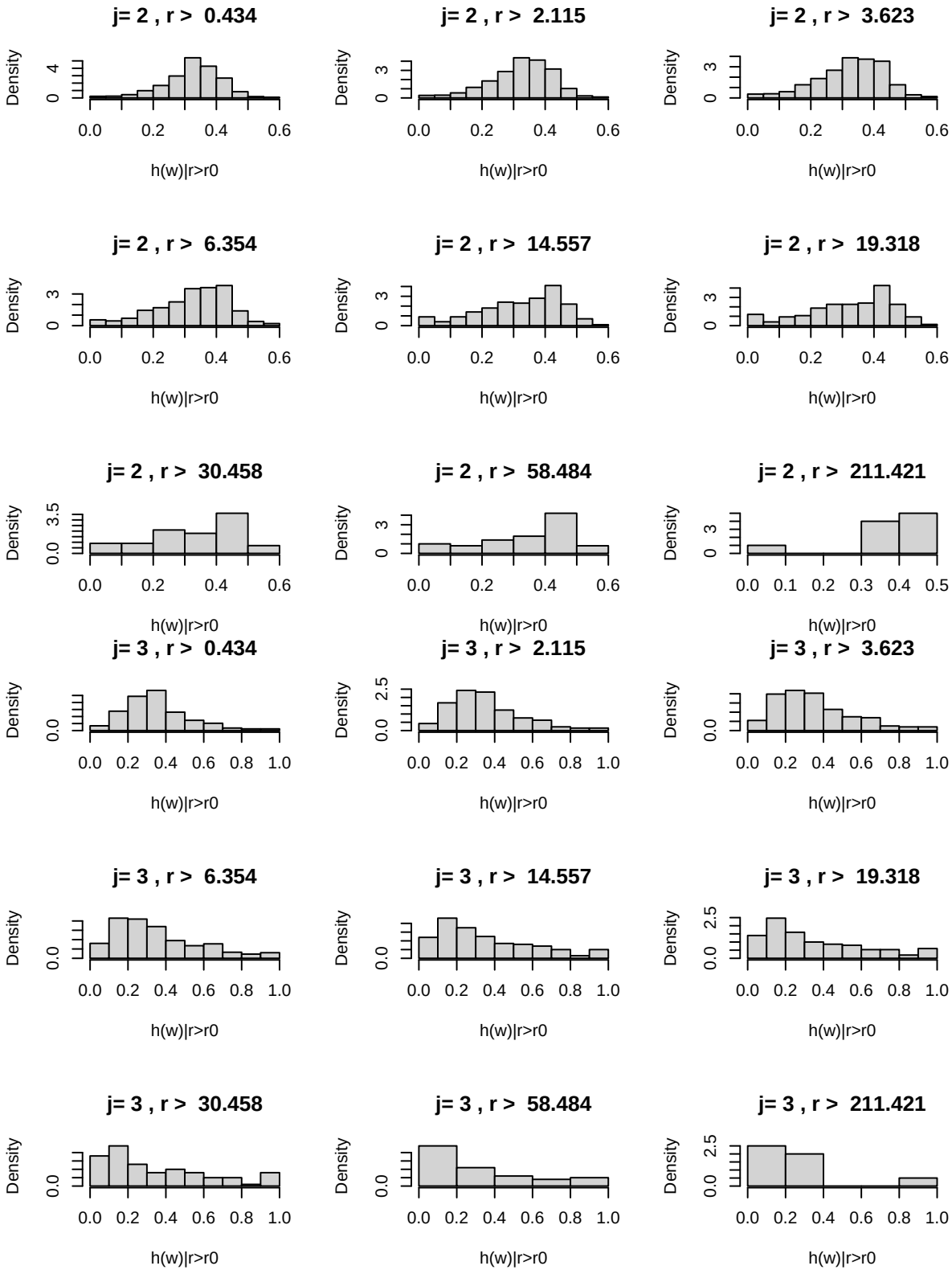
c) Similarly to question b) from Exercise 1, we need to identify a threshold value r_0 such that $\omega \perp r \mid r > r_0$. Produce histograms of $\omega \perp r \mid r > r_0$ for r_0 defined by the observed 0 (minimum), 0.2, 0.4, 0.6, 0.8, 0.85, 0.9, 0.95, and 0.99 quantiles of r. Hence choose a suitable threshold, r_0.

Solution:

```
q <- c(0,0.2,0.4,0.6,0.8,0.85,0.9,0.95,0.99)
r0 <- quantile(r,q)
for(j in 1:3){
  par(mfrow=c(3,3))
  for (i in 1:length(r0)){
```

```
    hist(w[r>r0[i],j], main=paste("j=",j,", r > ",round(r0[i],3)),
         probability=T, xlab="h(w)|r>r0")
  }
}
```





Unlike Question 1 with simulated data the dependence exhibited in the (pairwise) angular components doesn't vary very much with increasing r_0. Possibly there are several choices that could be made here. The one we make is that the histogram of the first currency (pair)

is stable from the $i = 3$ plot, and the other two from the $i = 5$ plot. In combination, this suggests that ($i = 5$) $r_0 = 14.557$, the 80th quantile of the observed radial components may be sufficient. (This is an assumption!) Under this assumption, we have 200 data points for analysis.

```
i <- 5 # choosen threshold
ind <- r>r0[i] # 200 observations
```

d) Using the approximate likelihood (`fExtDep` in `ExtremalDep`) with the observed $\omega \mid r > r_0$, fit both extremal-t and Hüsler-Reiss models in the region above the threshold, r_0. For each bivariate pair plot the empirical and fitted 2D angular densities (use the `dExtDep` function, divided by two!) by extracting the relevant parameters from the 3D model fit. (Note: these will be approximations to the 2D model fits . . .). Also plot the fitted 3D angular densities and superimpose the observed data (using `plot_ExtDep`). Assess the quality of the two model fits.

Note: You may need to be careful with your starting points when fitting the model. Also, specify $c = 0$ for the extremal-t model.

Solution:

```
library(ExtremalDep)
ddat <- w[ind,]
fit.et <- fExtDep(model="ET", data=ddat, par.start=c(rep(0.5,3),2), trace=T)
```

```
## initial  value -289.383931
## iter  10 value -669.965609
## final  value -675.229595
## converged
```

```
#c(fit.et$par, fit.et$TIC)
fit.hr <- fExtDep(model="HR", data=ddat, par.start=rep(2.5,3), trace=T)
```

```
## initial  value 777.227002
## iter  10 value -668.877380
## final  value -671.035739
## converged
```

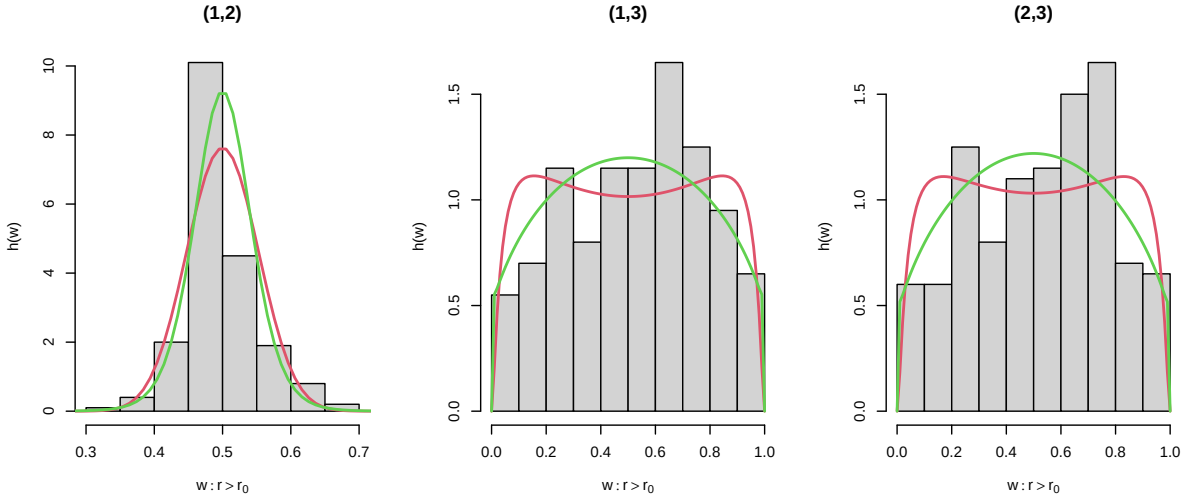
```
#c(fit.hr$par, fit.hr$TIC)
```

```
xx <- simplex(d=2, n=100)
par(mfrow=c(1,3))
hist(w[ind,1]/(w[ind,1]+w[ind,2]), probability=T, ylab="h(w)",
     xlab=expression(w : r>r[0]), main="(1,2)")
lines(xx[,1], apply(xx, 1, dExtDep, model="HR", par=fit.hr$par[1])/2, lwd=2, col=2)
lines(xx[,1], apply(xx, 1, dExtDep, model="ET", par=fit.et$par[c(1,4)], c=0)/2, lwd=2, col

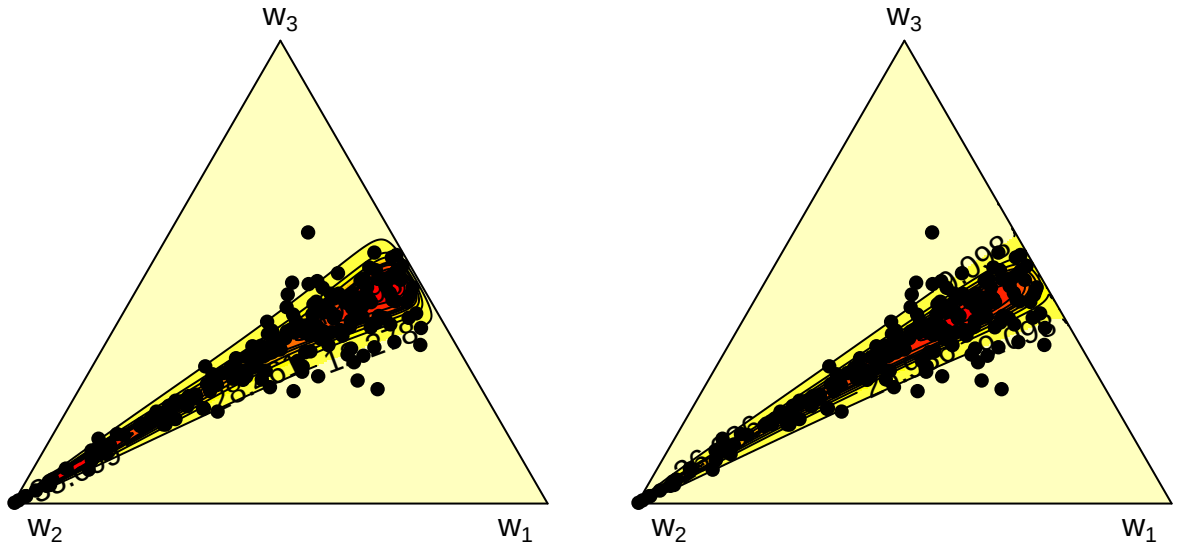
# Alternatives
# plot_ExtDep(model="HR", par=fit.hr$par[1], style="hist",
#             data=w[ind,1:2]/rowSums(w[ind,1:2]), log=FALSE)
# plot_ExtDep(model="HR", par=fit.hr$par[1], style="ticks",
#             data=w[ind,1:2]/rowSums(w[ind,1:2]), log=FALSE)
```

```
hist(w[ind,1]/(w[ind,1]+w[ind,3]), probability=T, ylab="h(w)",
     xlab=expression(w : r>r[0]), main="(1,3)")
lines(xx[,1], apply(xx, 1, dExtDep, model="HR", par=fit.hr$par[2])/2, lwd=2, col=2)
lines(xx[,1], apply(xx, 1, dExtDep, model="ET", par=fit.et$par[c(2,4)], c=0)/2, lwd=2, col

hist(w[ind,2]/(w[ind,2]+w[ind,3]), probability=T, ylab="h(w)",
     xlab=expression(w : r>r[0]), main="(2,3)")
lines(xx[,1], apply(xx, 1, dExtDep, model="HR", par=fit.hr$par[3])/2, lwd=2, col=2)
lines(xx[,1], apply(xx, 1, dExtDep, model="ET", par=fit.et$par[3:4], c=0)/2, lwd=2, col=3)
```




```
par(mfrow=c(1,2), mar=c(0, 0.4, 0.4, 0.4))
plot_ExtDep(model="HR", par=fit.hr$par, data=w[ind,], log=FALSE)
plot_ExtDep(model="ET", par=fit.et$par, data=w[ind,], log=FALSE)
```



For the (approximate) 2D angular densities, both extremal-t (green) and Hüsler-Reiss (red) models track the data reasonably well, within their parametric constraints. The extremal-t provides greater dependence in the $(1, 2)$ pair than the Hüsler-Reiss, and has more of a $\cup$ shape for the other two pairs, in contrast to the Hüsler-Reiss which, while forcibly symmetric,

is more uniform over all angles.
For the 3D angular density the fitted extremal-t model tracks the observed data fairly well.
For the Hüsler-Reiss model, the plot failed (maybe some coding issue).

e) Compute and interpret estimates of the extremal coefficient, θ, and the coefficient of upper tail dependence, χ. Do these estimates make sense? How do they compare to the estimates obtained in the Week 8 Lab when we analysed these data via block maxima?

Solution:

```
theta.hr <- index.ExtDep(object="extremal", model="HR", par=fit.hr$par)
theta.hr
```

```
## [1] 1.492347
```

```
theta.et <- index.ExtDep(object="extremal", model="ET", par=fit.et$par)
theta.et
```

```
## [1] 1.496913
```

```
chi.hr <- pExtDep(q=rep(1,3), type="upper", model="HR", par=fit.hr$par)
chi.hr
```

```
## [1] 0.4674614
```

```
chi.et <- pExtDep(q=rep(1,3), type="upper", model="ET", par=fit.et$par)
chi.et
```

```
## [1] 0.4859499
```

The estimates of the extremal coefficient θ are not to dissimilar from those obtained in the week 8 Lab when analysing the block (monthly; $n = 21$) maxima.
The interpretation and intuition remains the same: out of the 3 variables, two are very highly correlated, and the third is positively correlated with the others. The former will reduce our estimate of θ from 3 to slightly above 2, and the latter will reduce it further by some amount. As such, approximately $\theta \approx 1.5$ independent variables out of 3 seems quite credible.
The estimates of χ, which is the probability of lying in the upper tail region are around 0.46–0.49. Given that we have 3 dimensions, this is indicating a reasonable amount of dependence in the extremes, which agrees with what we know about the dataset.

Exercise 3

Plot the 2D and 3D angular density of the asymmetric logistic model for various values of the model's parameters. In two dimensions the angular density is given by:

$$\begin{aligned} h(\omega) &= (\alpha-1)(\beta_1\beta_2)^\alpha(\omega(1-\omega))^{\alpha-2}\left\{(\beta_1(1-\omega))^\alpha + (\beta_2\omega)^\alpha\right\}^{1/\alpha-2} \\ \mathbb{P}(\{0\}) &= 1-\beta_1 \\ \mathbb{P}(\{1\}) &= 1-\beta_2 \end{aligned}$$

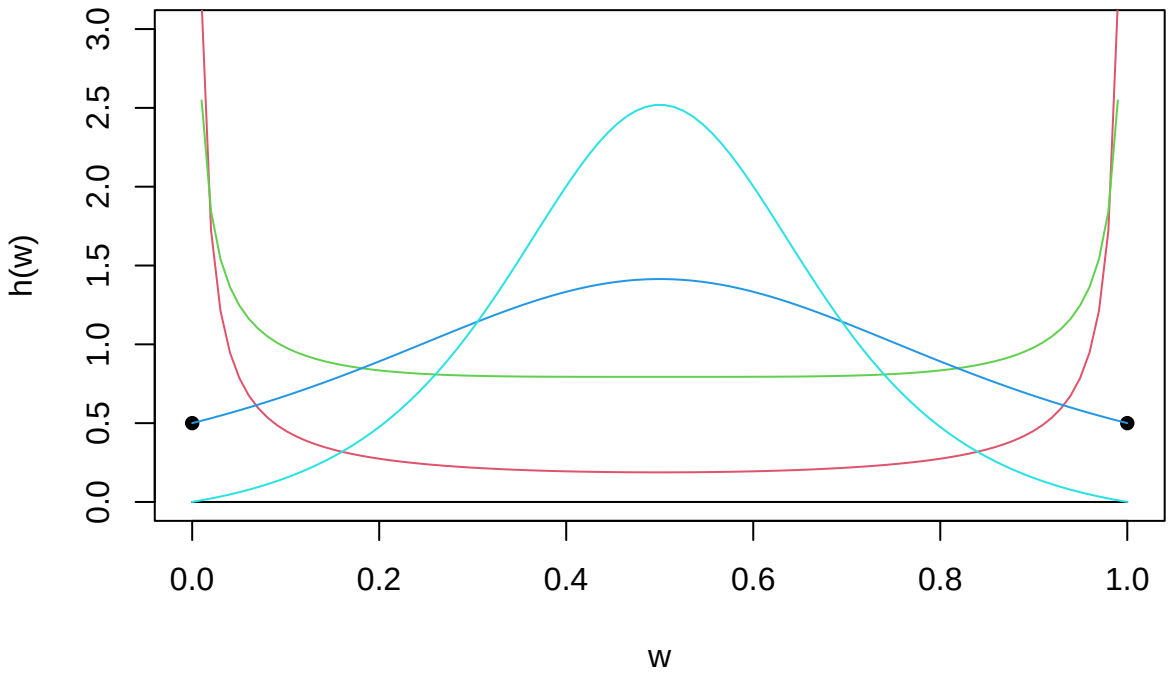
with parameters $\theta = (\alpha, \beta_1, \beta_2)$. In 3 dimensions the interior portion of the angular density is available through `plot_ExtDep` from `ExtremalDep`. What condition is needed to have a symmetric

dependence structure? What impact does each parameter have on the dependence structure? Can we obtain complete dependence and independence?

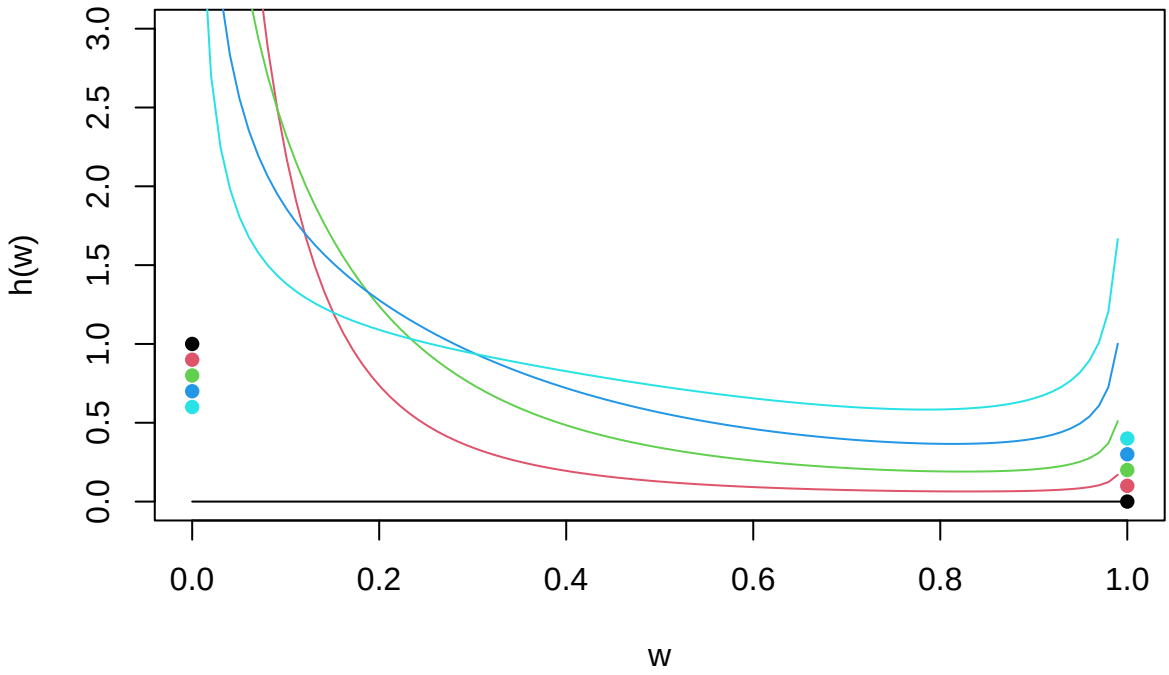
Solution:

```
xx <- simplex(d=2, n=100)

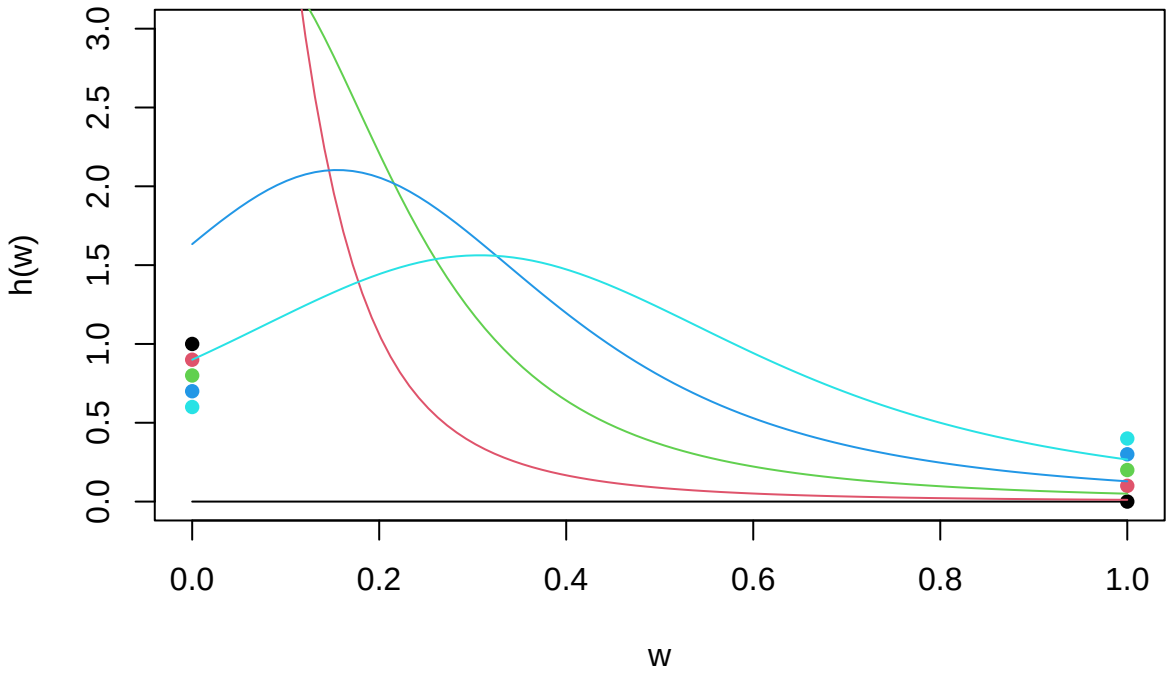
# Symmetric - vary alpha
plot(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1, 0.5, 0.95), c=0),
     type="l", ylab="h(w)", xlab="w", ylim=c(0,3))
points(xx[c(1,100),1], c(1-0.5, 1-0.5), pch=16)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1.1, 0.5, 0.5), c=0), col=2)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1.5, 0.5, 0.5), c=0), col=3)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(2.0, 0.5, 0.5), c=0), col=4)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(3.0, 0.5, 0.5), c=0), col=5)
```




```
# Alpha = weak dependence, vary asymmetry
plot(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1.5, 0, 1), c=0),
     type="l", ylab="h(w)", xlab="w", ylim=c(0,3))
points(xx[c(1,100),1], c(1-0, 1-1), pch=16)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1.5, 0.1, 0.9), c=0), col=2)
points(xx[c(1,100),1], c(1-0.1, 1-0.9), pch=16, col=2)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1.5, 0.2, 0.8), c=0), col=3)
points(xx[c(1,100),1], c(1-0.2, 1-0.8), pch=16, col=3)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1.5, 0.3, 0.7), c=0), col=4)
points(xx[c(1,100),1], c(1-0.3, 1-0.7), pch=16, col=4)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(1.5, 0.4, 0.6), c=0), col=5)
points(xx[c(1,100),1], c(1-0.4, 1-0.6), pch=16, col=5)
```

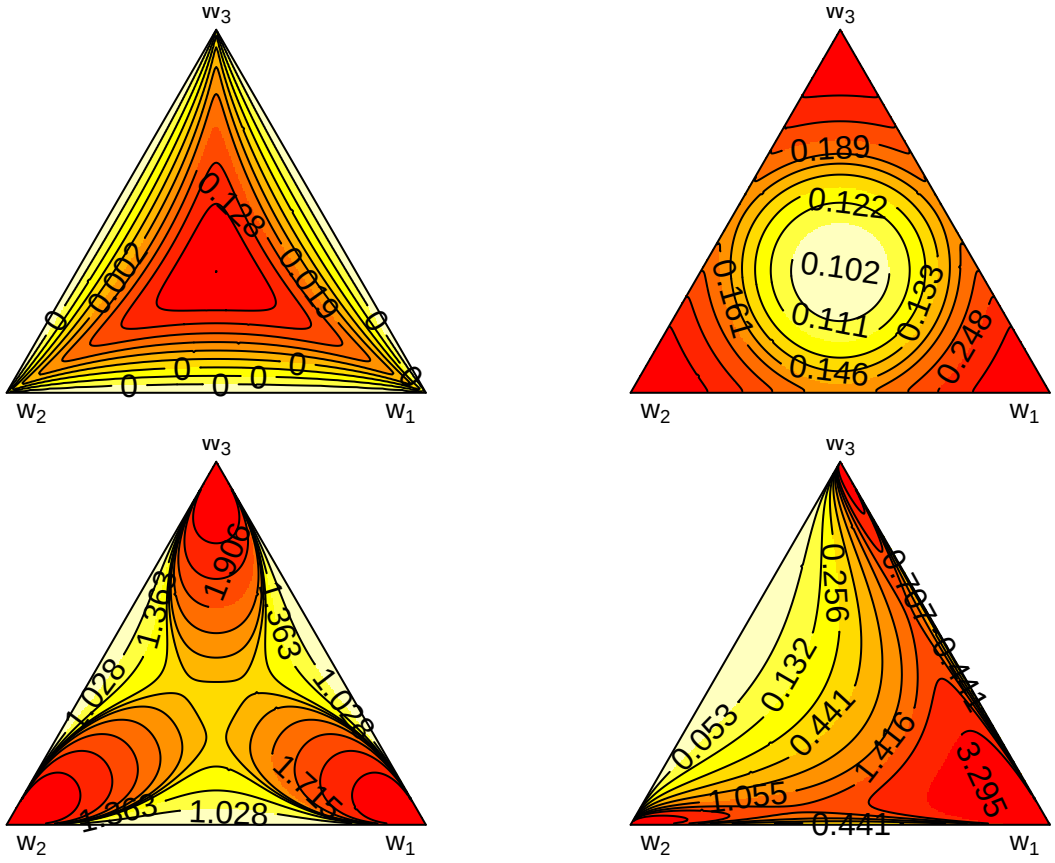



```
# Alpha = stronger dependence, vary asymmetry
plot(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(2, 0, 1), c=0),
     type="l", ylab="h(w)", xlab="w", ylim=c(0,3))
points(xx[c(1,100),1], c(1-0, 1-1), pch=16)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(2, 0.1, 0.9), c=0), col=2)
points(xx[c(1,100),1], c(1-0.1, 1-0.9), pch=16, col=2)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(2, 0.2, 0.8), c=0), col=3)
points(xx[c(1,100),1], c(1-0.2, 1-0.8), pch=16, col=3)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(2, 0.3, 0.7), c=0), col=4)
points(xx[c(1,100),1], c(1-0.3, 1-0.7), pch=16, col=4)
lines(xx[,1], apply(xx, 1, dExtDep, model="AL", par=c(2, 0.4, 0.6), c=0), col=5)
points(xx[c(1,100),1], c(1-0.4, 1-0.6), pch=16, col=5)
```



In three dimensions, the model has 13 parameters. We write them as $(1, 1, 1, \alpha, 0, 0, 0, 0, 0, 0, \beta_1, \beta_2, \beta_3)$ and in the code chunk below, we only need to specify `par=c(alpha, beta1, beta2, beta3)`.

```
par(mfrow=c(2,2), mar=c(0, 0.4, 0.4, 0.4))
plot_ExtDep(model="AL", par=c(5.75,0.5,0.5,0.5), log=FALSE)
plot_ExtDep(model="AL", par=c(1.01,0.9,0.9,0.9), log=FALSE)
plot_ExtDep(model="AL", par=c(1.25,0.5,0.5,0.5), log=FALSE)
plot_ExtDep(model="AL", par=c(1.40,0.7,0.15,0.15), log=FALSE)
```



For the 2D plots (First plot): keeping $\beta_1 = \beta_2$ and setting $\alpha = 1$ gives independence (black line). Increasing α increases the dependence from point masses on the vertices, through a $\cup$ shape to a $\cap$ shape, and (in the limit $\alpha \to \infty$) a point mass on $\{0.5\}$.
(Second plot): Holding $\alpha = 1.5$ at weak dependence, varying β_1 (and setting $\beta_2 = 1 - \beta_1$) from 0 (independence) to 0.4 weights the $\cup$ shape to varying degrees in the direction of the $\{0\}$ vertex.
(Third plot): Holding $\alpha = 2$ at stronger dependence, varying β_1 (and setting $\beta_2 = 1 - \beta_1$) from 0 (independence) to 0.4 weights the $\cap$ shape to varying degrees in the direction of the $\{0\}$ vertex.
We can see that we get independence if $\alpha = 1$ or if either $\beta_1 = 0$ or $\beta_2 = 0$. We can get complete dependence if $\alpha \to \infty$ and if $\beta_1 \neq 0$ and $\beta_2 \neq 0$.

For the 3D plots (top left): Large α and equal β_i's gives strong symmetric dependence. (top right): Small α and equal β_i's gives strong independence with $\cup$ shapes on the 2D margins. (bottom left): Similar $\cup$ shapes on the 2D margins but with an additional peak at $w = 0.5$ resulting from a more structured dependence structure. (bottom right): Asymmetry in the β_i's results in asymmetry in the dependence structure.
As before, $\alpha = 1$ or any $\beta_i = 0$ will give complete independence. $\alpha \to \infty$ gives complete dependence.